

A Local Gateway for Headless Claude and Codex

M Walker 10 October 2026

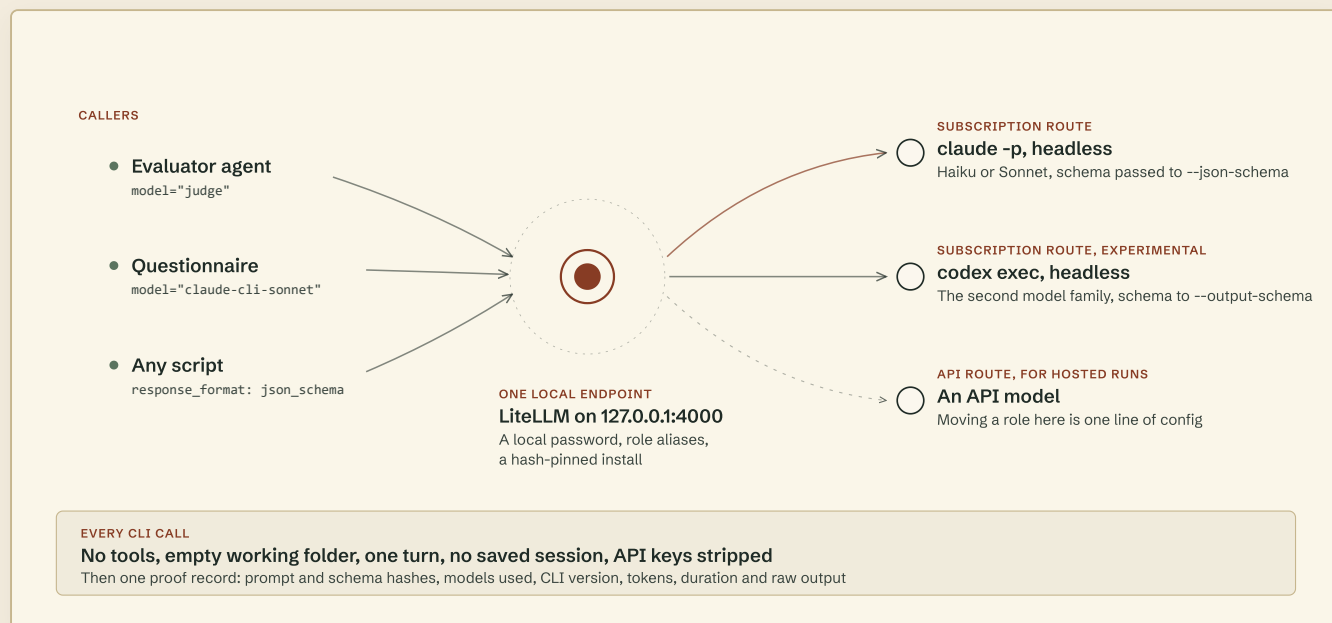
I built a personal gateway that gives every agent I write one OpenAI-compatible endpoint. Role aliases decide whether a call goes to headless Claude Code, headless Codex or an API model, and every call leaves a proof record.

◇ At a glance

The gateway is LiteLLM 1.104.0 on 127.0.0.1, hash pinned, with two custom providers I wrote: one runs `claude -p` and the other `codex exec`, both on my own logins. Every model call in my Offerings Atlas went through it, 302 of them on 10 October 2026 alone, at no marginal cost.

◇ Why

I wanted the agents in my projects to call models the way production code would: one API, structured output, a role name instead of a model name. I did not want to pay API rates for every experiment while I build, and I did not want a coding assistant's tools, memory or instructions leaking into calls that are meant to be a clean judgement.



◇ How it works

Callers ask for a role such as `judge` . One line of configuration maps the role to a route, so moving the judge from Haiku on my subscription to an API model for a hosted run is a one-line change. A JSON schema in the request goes straight to the CLI's own schema flag, so the caller gets valid JSON back or a clear error.

Each CLI call is locked down:

- no tools and no MCP servers, with a permission mode that cannot prompt
- an empty temporary working folder, one turn and no saved session
- Claude Code's own system prompt replaced, so no memory, project instructions or skills reach the model
- API keys stripped from the child process, so the call bills the subscription and not an account

Each call then writes one proof record: hashes of the system prompt, the prompt and the schema, the model requested and the models actually used, the CLI version and login method, tokens, duration, the tool's notional cost and the raw output.

◇ In use

Measure	Value
Calls through the gateway on 10 October 2026	302
Sonnet judge call on a real knowledge base cell	about 6 seconds and 14,500 input tokens
Haiku judge call on a test case	about 5 seconds and 2,800 input tokens

Two practical lessons came out of running it hard. Several Claude sessions refreshing the same login at once can make a call fail for a few seconds, so callers retry with backoff. And Codex loads its home folder's instructions even with user configuration ignored, so the gateway supports a separate Codex home with its own login.

◇ Limits

- It is personal tooling for my own machine. The subscription routes bind to 127.0.0.1 and serve only me.
- Hosted or shared use should move to API keys, which the configuration already supports.
- OpenAI recommends API keys for automation, so I treat the Codex route as experimental.

Figures come from the gateway's own proof log as of 10 October 2026.